

Fundamentals and Benefits of CI/CD

Anh Le - AnhLD41





The world before CI/CD

Everything requires human intervention which would leads to a huge amount of human errors





Warning signs

Some issues we might struggle with

- Investing **more time** in a release cycle than delivering value
- Going through integration hell every time we finish a feature
- **Code gets lost** because of botched merges
- Unit test suite hasn't been green in ages
- Deployments contribute to **schedule slip**
- Friction between ops and development departments
- **Only one engineer** can deploy a system
- ***Deployments are not cause for celebration***

CI/CD



- **No more** manual deploying to environments
- **No more** modifying environment settings in GUI's
- **No more** neglecting the unit tests
- **No more** leaving broken code in place
- Requires a high level of discipline
- Requires additional skills to maintain and extend automation



Fundamentals

Continuous Integration (CI)

This is an automation process for developers to have our code automatically built, tested, and merged to a shared repository.

- Compile
- Unit Test
- Static Analysis
- Dependency vulnerability testing
- Store artifact

Continuous Deployment (CD)

This is the way to automatically release a developer's changes from the repository to production for customer to test and verify as a real product.

- Creating infrastructure
- Provisioning servers
- Copying files
- Promoting to production
- Smoke testing
- Rollbacks



Benefits

- Less developer time on issues from new developer code
- Less bugs in production and less time in testing
- Less human error, Faster deployments
- Less time to market
- Reduced downtime from a deploy-related crash or major bug
- New value-generating features released more quickly



- Reduce cost
- Avoid code
- Increase revenue